Technische Universität Chemnitz
Prof. Dr. R. Herzog, M. Winkler

Chemnitz, 29. November 2019
Abgabe am 11. Dezember 2019

Grundlagen der Optimierung

Übung 8

Aufgabe 32: Duales LP für verschiedene Beispiele

(a) Zeige, dass sich das duale vom dualen eines LP in Normalform wieder als das primale Problem auffassen lässt.

(b) Zeige, dass sich die rechten linearen Programme jeweils als das duale der linken linearen Programme auffassen lassen.

(a)

$$\begin{array}{lll} \max & c^\top x \\ \text{sodass} & Ax \leq b \\ & x \geq 0 \end{array} \qquad \begin{array}{lll} \min & b^\top \lambda \\ \text{sodass} & A^\top \lambda \geq c \\ & \lambda \geq 0 \end{array}$$

(b)

$$\begin{array}{lll} \min & c^\top x \\ \text{sodass} & Ax \geq b \\ & x \text{ frei} \end{array} \qquad \begin{array}{lll} \max & b^\top \lambda \\ \text{sodass} & A^\top \lambda = c \\ & \lambda \geq 0 \end{array}$$

(c)

$$\begin{array}{lll} \max & c^\top x \\ \text{sodass} & Ax = b \\ & \ell \leq x \leq u \end{array} \qquad \begin{array}{lll} \min & (A\ell - b)^\top \lambda_1 + (u - \ell)^\top \lambda_2 + c^\top \ell \\ \text{sodass} & A^\top \lambda_1 - \lambda_2 \leq -c \\ & \lambda_1 \text{ frei},\ \lambda_2 \geq 0 \end{array}$$

Hinweis: Überführe zunächst in Normalform.

Aufgabe 33: Beispiele für LPs mit (un-)zulässigen und (un-)beschränkten primalen/dualen Problemen

(a) Gib ein lineares Programm an, bei dem sowohl das primale als auch das duale Problem unzulässig ist (vgl. Satz 8.7, Fall (II)).

(b) Gib ein lineares Programm an, bei dem das primale zulässig und unbeschränkt ist und das zugehörige duale Problem unzulässig ist (vgl. Satz 8.7, Fall (III)).

Aufgabe 34: Gemischte Erweiterung eines Matrixspiels

Sei $A = (a_{ij}) \in \mathbb{R}^{m\times n}$ eine (Gewinn-)matrix. Die Spieler P_1 und P_2 spielen folgendes Spiel: P_1 würfelt eine Spalte von A aus, d.h. eine Zahl $j \in \{1,\dots,n\}$, und P_2 würfelt eine Zeile $i \in \{1,\dots,m\}$. Dann zahlt P_2 an P_1 den Wert a_{ij}. (Falls $a_{ij} < 0$, so zahlt P_1 an P_2 den Wert $-a_{ij}$.) Beide Spieler können vorher (geheim, d.h. unabhängig voneinander) die Wahrscheinlichkeitsverteilung festlegen, mit der gewürfelt wird. P_1 wählt ein $x \in \mathbb{R}^n$ mit $x \geq 0$, $\sum_{j=1}^n x_j = 1$ und würfelt im Spiel die Spalte j mit Wahrscheinlichkeit x_j. Analog wählt P_2 ein $y \in \mathbb{R}^m$ mit $y \geq 0$, $\sum_{i=1}^m y_i = 1$.

(a) Angenommen, P_1 kennt durch einen Spion die Wahrscheinlichkeitsverteilung $\bar{y} \in \mathbb{R}^m$, $\bar{y} \geq 0$, $\sum_{i=1}^m \bar{y}_i = 1$, mit der P_2 die Zeilen auswählen wird. Zeige, dass P_1 den Erwartungswert seines Gewinns maximiert, wenn er eine optimale Lösung des linearen Programms

$$\begin{array}{ll} \max & \bar{y}^\top Ax \quad \text{über } x \in \mathbb{R}^n \\ \text{sodass} & e^\top x = 1 \\ & x \geq 0 \end{array}$$

als seine Wahrscheinlichkeitsverteilung zur Spaltenwahl nutzt. Dabei sei $e := (1,\dots,1)^\top$. Wie sieht die Menge der Optimallösungen dieser Aufgabe aus?

(b) Der Spion hat die Seiten gewechselt und steht nun P_2 zur Verfügung. Zeige, dass unter diesen Umständen P_1 den Erwartungswert seines Gewinns maximiert, wenn er den x-Anteil einer optimalen Lösung des linearen Programms

$$\begin{array}{ll} \max & v \quad \text{über } x \in \mathbb{R}^n,\ v \in \mathbb{R} \\ \text{sodass} & v \leq e_i^\top Ax, \quad i = 1,\dots,m \\ & e^\top x = 1 \\ & x \geq 0,\ v \text{ frei} \end{array}$$

als Wahrscheinlichkeitsverteilung zur Spaltenwahl nutzt. Dabei sei e_i der i-te Einheitsvektor.

(c) Kann P_1 den erwarteten Gewinn aus (b) verbessern, wenn er P_2 den Spion wieder abwirbt? Zeige, dass dies nicht der Fall ist, wenn P_2 für seine Wahrscheinlichkeitsverteilung ein zu (b) analoges lineares Programm löst. (Tipp: starke Dualität)

Hausaufgabe 19: Implementierung von Phase I

Implementiere „Phase I“ für ein lineares Programm in Normalform in `Matlab`. Erstelle dazu eine Datei `simplex_phaseI.m` und verwende

```
function [x,basis] = simplex_phaseI(A,b,c,pricing)
```

als erste Zeile. Dabei sind `A,b,c` die Daten des linearen Programms

$$\begin{aligned} \min \quad & c^\top x \\ \text{sodass} \quad & Ax = b \\ & \;\; x \geq 0 \end{aligned}$$

und `pricing` gibt die Auswahlstrategie an. Rückgabewerte sind ein für Phase II zulässiger Basisvektor `x` und die zugehörige Basis `basis`. Auch der Fall, dass kein zulässiger Punkt existiert, ist abzufangen. Teste den Algorithmus an dem in Übung 7, Aufgabe 28 angegebenen Beispiel.

Hinweis: Verwende zur Lösung des Hilfsproblems die Implementierung des Simplex-Algorithmus aus Aufgabe Übung 7, Hausaufgabe 18. Falls diese Aufgabe nicht gelöst wurde kann auch die MATLAB-Funktion `linprog` bzw. die Octave-Funktion `glpk` verwendet werden.

Abgabe: Schicke die erzeugten `m`-Files `simplex_phaseI.m` und `test_phaseI.m` an `max.winkler@mathematik.tu-chemnitz.de` (Betreff: HA Grundlagen der Optimierung Übung 8).

Hausaufgabe 20: Würfel von Klee-Minty

Wir betrachten das lineare Problem

$$\begin{array}{rlllllllll}
\max & 2^{n-1}x_1 + 2^{n-2}x_2 + \ldots + 2x_{n-1} + x_n \\
\text{sodass} & x_1 & & & & & & & \leq & 5 \\
& 4x_1 & + & x_2 & & & & & \leq & 25 \\
& 8x_1 & + & 4x_2 & + & x_3 & & & \leq & 125 \\
& \vdots & & & & & & & & \vdots \\
& 2^n x_1 & + & 2^{n-1}x_2 & + & \ldots & + & 4x_{n-1} \; + \; x_n & \leq & 5^n \\
\text{und} & x \geq 0.
\end{array}$$

Löse das Problem für verschiedene n mit dem Simplex-Algorithmus in MATLAB. Verwende dabei die Indizes der eingeführten Schlupfvariablen als Startbasis und *negativste reduzierte Kosten* als Auswahlstrategie. Wie hängen die vom Simplex-Algorithmus benötigten Iterationen mit n zusammen?

Hinweis: Verwende zur Lösung des Hilfsproblems die Implementierung des Simplex-Algorithmus aus Aufgabe Übung 7, Hausaufgabe 18. Falls diese Aufgabe nicht gelöst wurde kann auch die MATLAB-Funktion `linprog` bzw. die Octave-Funktion `glpk` verwendet werden. Diese Funktionen verwenden allerdings ein primal-duales Simplex-Verfahren, welches bei diesem Problem deutlich weniger Iterationen benötigt um eine Lösung zu finden.

Abgabe: Schicke das erzeugte `m`-File `klee_minty.m` an `max.winkler@mathematik.tu-chemnitz.de` (Betreff: HA Grundlagen der Optimierung Übung 8).